

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A computer network comprising: - at least two processing nodes, each node having a processor on which one or more user processes are executed and a respective network interface; and a switching network which operatively connects the at least two processing nodes together, each network interface including a command processor and an addressable memory wherein the command processor of ~~said network interface~~ is configured to allocate ~~to a user process the exclusive right to~~ one or more segments of the addressable memory for the exclusive use of a user process in said network interface as one or more command queues mapped by said network interface to said allocated addressable memory.
2. (Original) A computer network as claimed in claim 1, wherein each one of said memory segments allocated as command queues is a contiguous series of memory addresses.
3. (Original) A computer network as claimed in claim 1, wherein each memory segment of the network interface memory allocated as a command queue has associated with it a queue descriptor which includes a user process identification.
4. (Original) A computer network as claimed in claim 3, wherein each queue descriptor includes an insert pointer identifying within the allocated memory

segment the memory address where new commands from the relevant user process are to be written.

5. (Original) A computer network as claimed in claim 3, wherein each queue descriptor includes a completed pointer identifying within the allocated memory segment the memory address of the end of the most recent completed command.

6. (Original) A computer network as claimed in claim 1, wherein the network interface includes a scheduler configured to identify and schedule any active command queues in the network interface memory.

7. (Original) A computer network as claimed in claim 6, wherein the scheduler has two or more run queues with at least one of the run queues being denominated a high priority run queue and at least one other of the run queues being denominated a low priority queue.

8. (Original) A computer network as claimed in claim 1, wherein said network interface includes at least one of the following resources: a thread processor, short transaction engine and a DMA engine, and each command queue stored in said network interface memory has associated with it a corresponding one or more virtual resources.

9. (Currently amended) A network interface comprising a command processor and an addressable memory wherein the command processor of said network interface is configured to allocate ~~to a user process being executed on a processor with which the network interface is associated, the exclusive rights to~~ one or more segments of the addressable memory for the exclusive use of a user process in said

~~network interface~~ as one or more command queues mapped by said network interface to said allocated addressable memory.

10. (Original) A network interface as claimed in claim 9, wherein each one of said memory segments allocated as command queues is a contiguous series of memory addresses.

11. (Previously presented) A network interface as claimed in claim 9, wherein each memory segment of the network interface memory allocated as a command queue has associated with it a queue descriptor which includes a user process identification.

12. (Original) A network interface as claimed in claim 11, wherein each queue descriptor includes an insert pointer identifying within the allocated memory segment the memory address where new commands from the relevant user process are to be written.

13. (Original) A network interface as claimed in claim 11, wherein each queue descriptor includes a completed pointer identifying within the allocated memory segment the memory address of the end of the most recent completed command.

14. (Original) A network interface as claimed in claim 9, wherein the network interface includes a scheduler configured to identify and schedule any active command queues in the network interface memory.

15. (Original) A network interface as claimed in claim 14, wherein the scheduler has two or more run queues with at least one of the run queues being denominated

a high priority run queue and at least one other of the run queues being denominated a low priority queue.

16. (Original) A network interface as claimed in claim 9, wherein said network interface includes at least one of the following resources: a thread processor, short transaction engine and a DMA engine, and each command queue stored in said network interface memory has associated with it a corresponding one or more virtual resources.

17. (Currently amended) A method of storing and running commands issued by a processor having associated with it a network interface comprising a command processor and a network interface memory, comprising the steps of:

the network interface receiving a request for a command queue from a user process being executed on the processor;

in response to the request allocating ~~to the user process exclusive rights to a~~ memory segment of the network interface memory for the exclusive use by the user process as a command queue mapped to said allocated memory segment;

storing one or more commands associated with the user process in said command queue; and

running said commands in said command queue without further intervention from said processor.

18. (Previously presented) A method of claim 17, wherein in response to requests from a plurality of user processes being executed on the processor, a respective plurality of memory segments of the network interface memory are allocated by the network processor.

19. (Previously presented) A method of claim 17, further comprising the step of said network processor generating a queue descriptor, which includes a user process identification, for each allocated memory segment.

20. (Previously presented) A method of claim 17, wherein said network interface includes a scheduler and said method further comprises the step of generating a run queue of active command queues in said network interface memory each active command queue containing at least one command awaiting execution.

21. (Previously presented) A method of claim 20, wherein at least two run queues are generated including a high priority run queue and a low priority run queue.